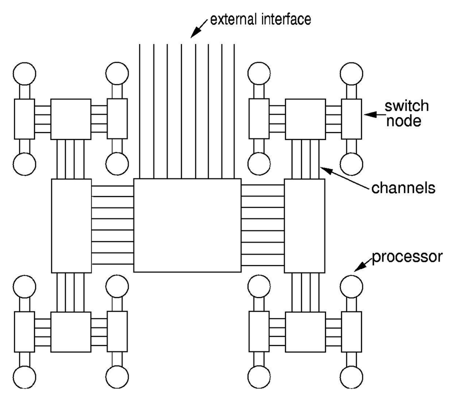
Fat-Tree Overlays for Scalable Multisource Multicast

http://www.aqualab.cs.northwestern.edu/projects/FatNemo.html
Contact: {sbirrer | fabianb}@cs.northwestern.edu

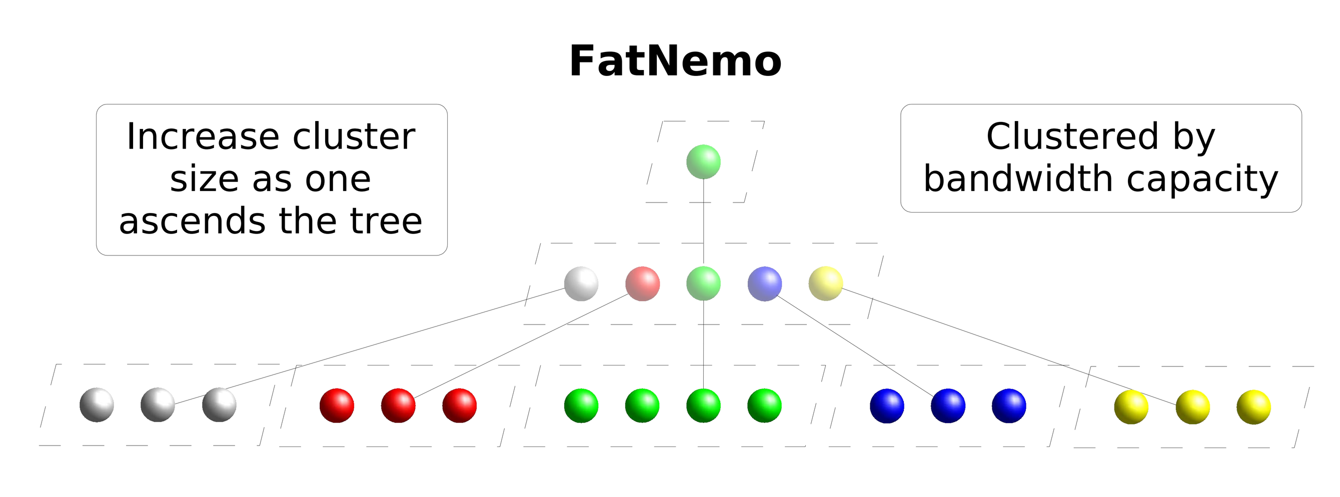
Stefan Birrer, Fabián E. Bustamante, Dong Lu, Peter Dinda and Yi Qiao
Department of Computer Science
Northwestern University

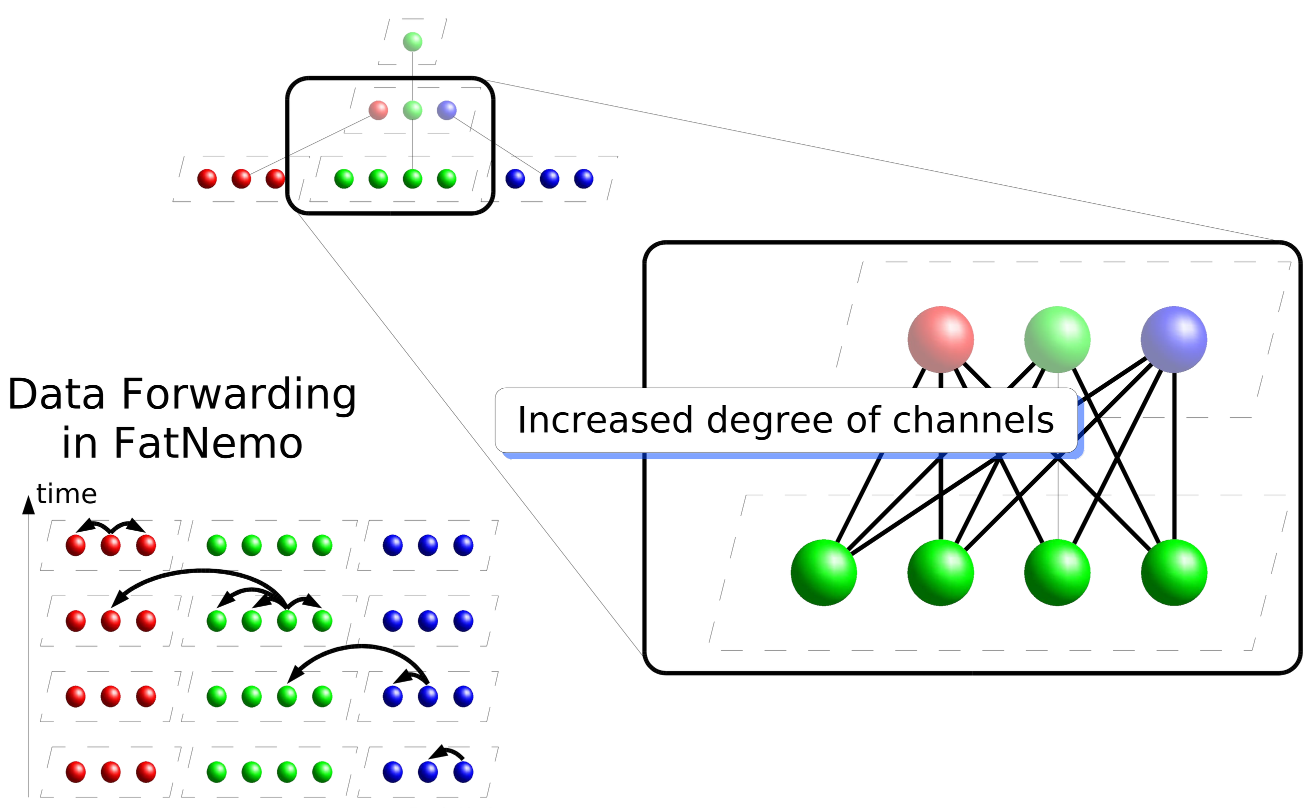
FatTrees for Overlay Multicast

Leiserson's fat-trees
- Minimal and scalable diameter
- Maximal and scalable bisection bandwidth

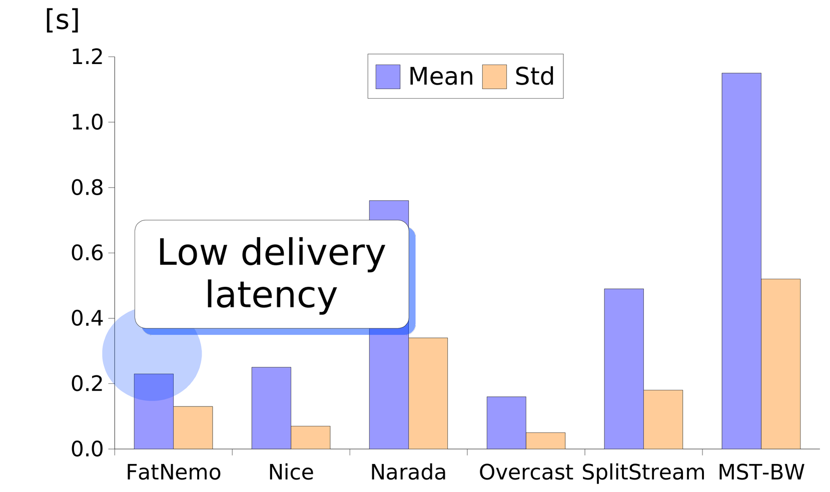


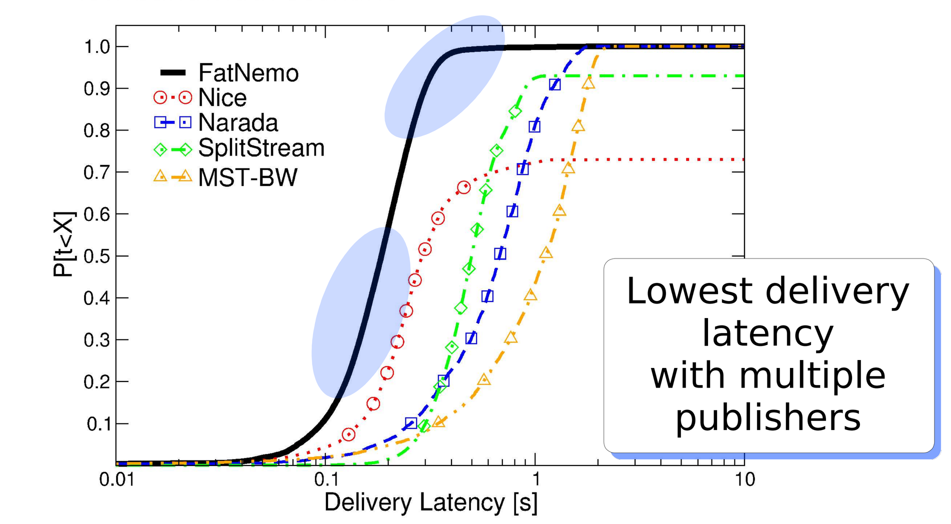
Resemble Leiserson's fat-trees on the overlay

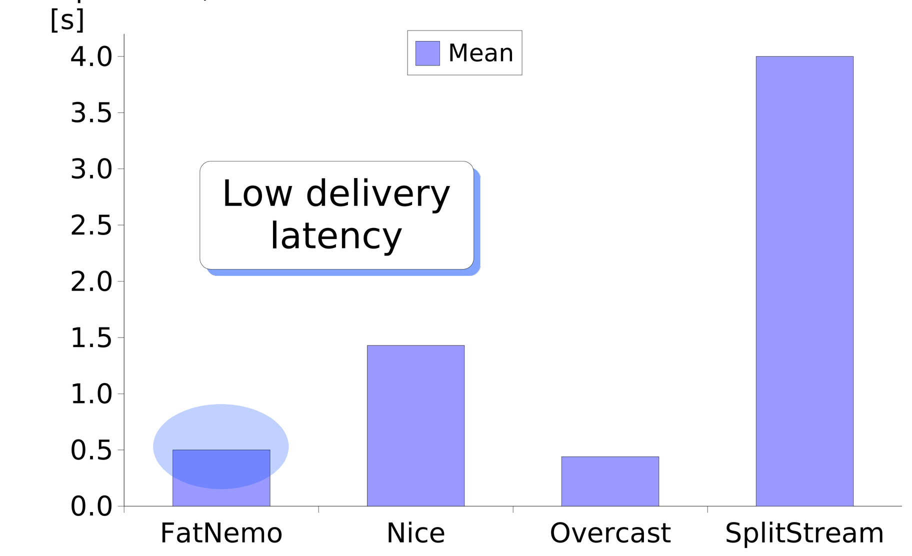




FatTree Performance







- **The need for efficient group communication**
  - **Multiplayer online gaming**
  - **Video conferencing**
  - **Cooperative Virtual Environments**
  - **Content distribution**

Our Goal

Enabling large scale multisource multicast applications.

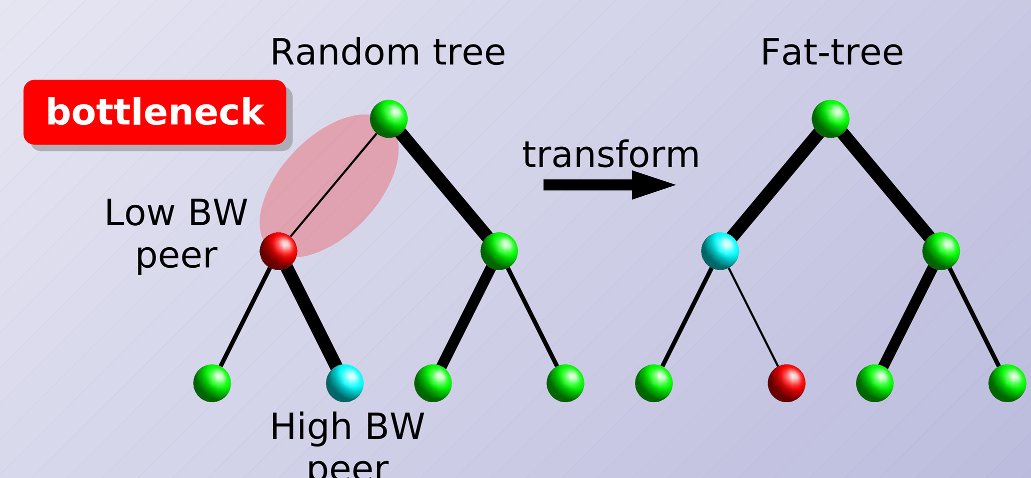
Our Approach

Fat Trees for Root Bottleneck Problem

We bypass the root bottleneck problem found in conventional tree-based systems by emulating Leiserson's fat-trees. The resulting overlay fat-trees have peers with higher bandwidth capacity located higher up in the hierarchy.

The Challenge

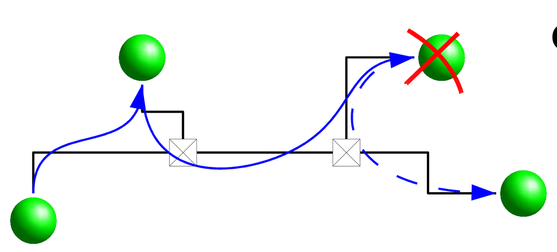
Handling highly transient populations and leverage/ respect heterogeneity.



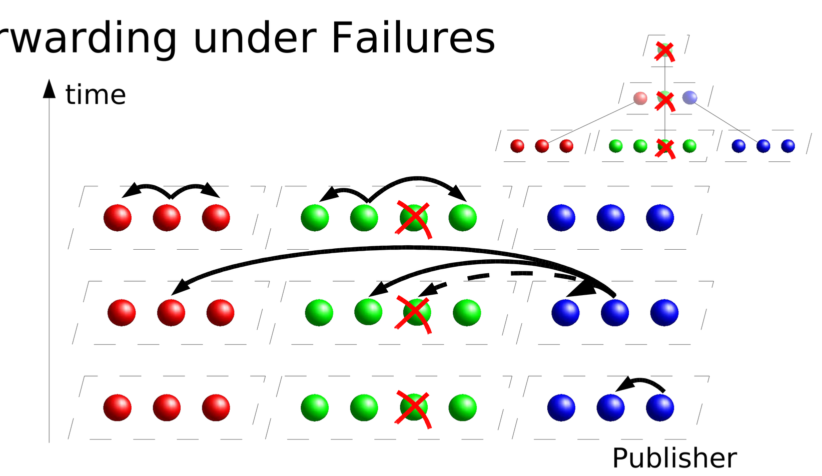
Two-metric Approach:
Sufficient bandwidth & minimized latency

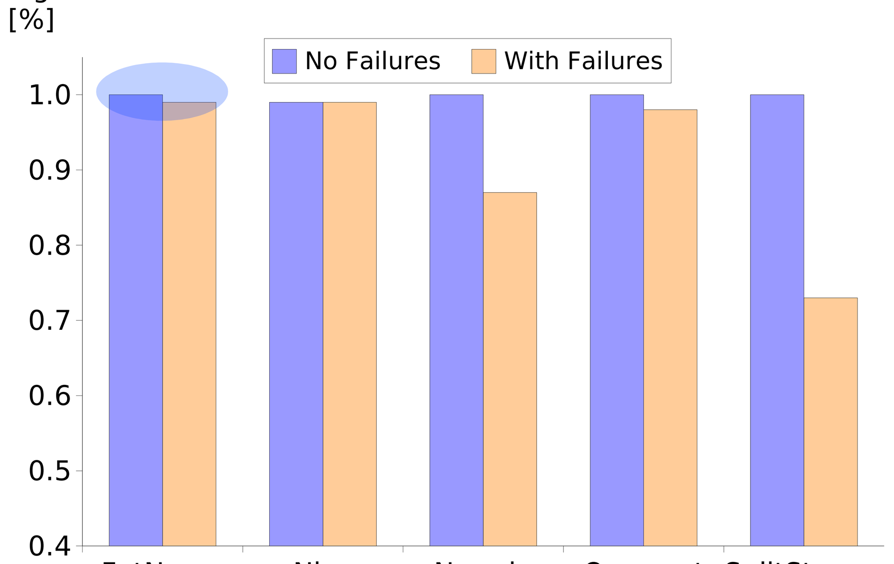
The overlay fat tree promotes the optimal peer based on the latency metric which also has *sufficient* bandwidth to the next higher layer. The algorithm is completely decentralized.

FatTree Resilience



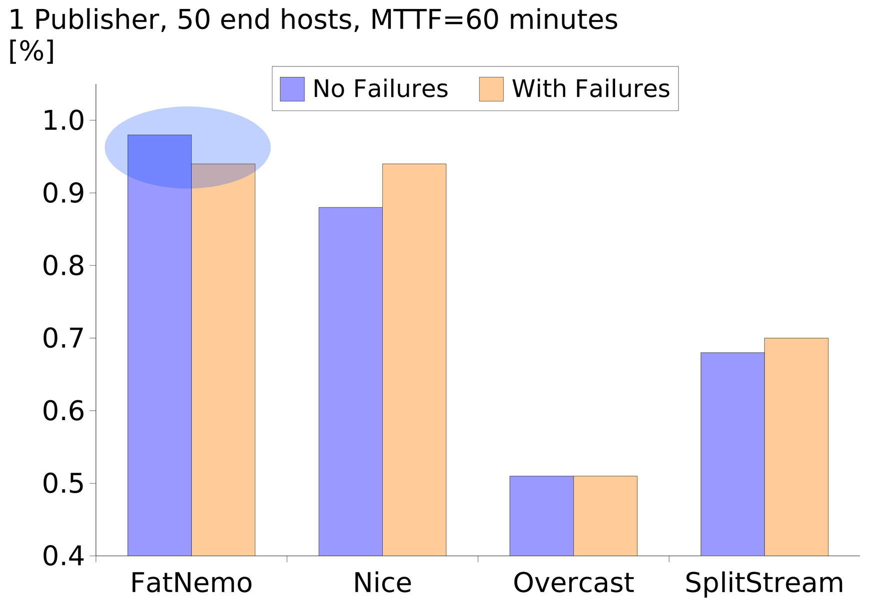
Conventional Trees
- Dependent on reliability of non-leave nodes
- Overlay nodes are highly transient Median session time 1 hr - 1 min.





Handles failures well

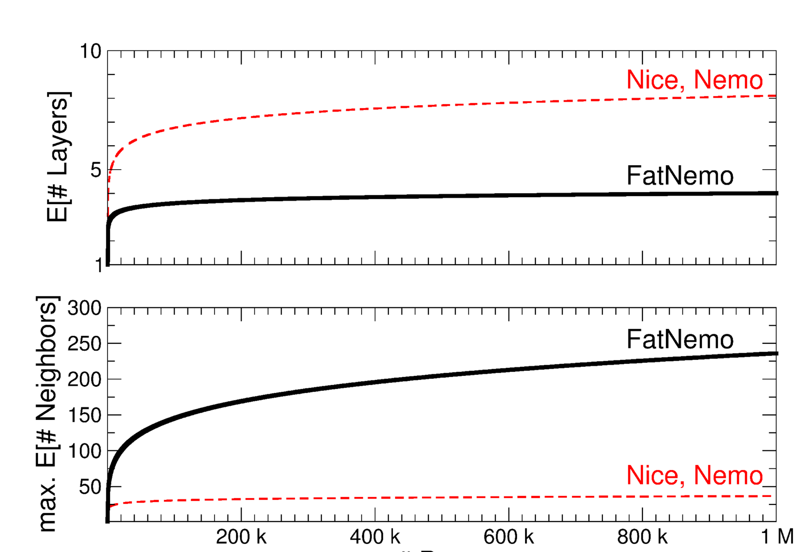
Nemo – Resilient P2P Multicast
Structural robustness through high path diversity
[MMCN'05]



FatTree Scalability

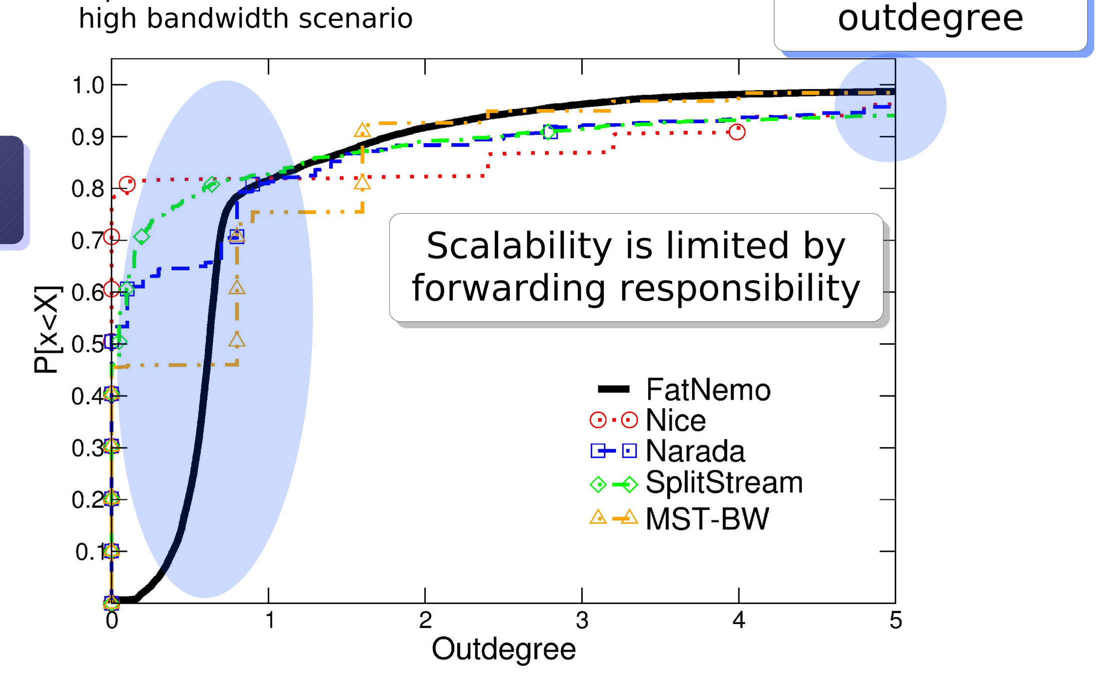
Scalability of tree-based protocols
cluster degree *d* of 3 with expected cluster size of 5.5 (Nice, Nemo)

Protocol	E[Layers]	E[Outdegree]
Nice	$\log_{5.5}(N)$	$4.5\,E[Layers] = 4.5\log_{5.5}(N)$
Nemo	$\log_{5.5}(N)$	$1.5\,E[Layers] = 1.5\log_{5.5}(N)$
FatNemo	$\approx \frac{-2.5+\sqrt{6.2+8.8\ln(N)}}{2.2}$	$< E[Layers] = \frac{-2.5+\sqrt{6.2+8.8\ln(N)}}{2.2}$



- Low expected number of layers
- Num. of neighbors grows only slowly

FatNemo is highly scalable



Ideally most peers should work in this region

- Outdegree defines the forwarding load of the peers
- High outdegree leads to overloaded peers